

# Online Research @ Cardiff

This is an Open Access document downloaded from ORCA, Cardiff University's institutional repository: <https://orca.cardiff.ac.uk/id/eprint/112882/>

This is the author's version of a work that was submitted to / accepted for publication.

Citation for final published version:

Gutiérrez, P.A., Salcedo-Sanz, S., Hervás-Martínez, C., Carro-Calvo, L., Sanchez-Monedero, Javier ORCID: <https://orcid.org/0000-0001-8649-1709> and Prieto, L. 2013. Ordinal and nominal classification of wind speed from synoptic pressure patterns. Engineering Applications of Artificial Intelligence 26 (3) , pp. 1008-1015. 10.1016/j.engappai.2012.10.018 file

Publishers page: <http://dx.doi.org/10.1016/j.engappai.2012.10.018>  
<<http://dx.doi.org/10.1016/j.engappai.2012.10.018>>

Please note:

Changes made as a result of publishing processes such as copy-editing, formatting and page numbers may not be reflected in this version. For the definitive version of this publication, please refer to the published source. You are advised to consult the publisher's version if you wish to cite this paper.

This version is being made available in accordance with publisher policies.

See

<http://orca.cf.ac.uk/policies.html> for usage policies. Copyright and moral rights for publications made available in ORCA are retained by the copyright holders.



# Ordinal and nominal classification of wind speed from synoptic pressure patterns<sup>☆</sup>

P. A. Gutiérrez<sup>\*,a</sup>, S. Salcedo-Sanz<sup>b</sup>, C. Hervás-Martínez<sup>a</sup>, L. Carro-Calvo<sup>b</sup>,  
J. Sánchez-Monedero<sup>a</sup>, L. Prieto<sup>c</sup>

<sup>a</sup>*Department of Computer Science and Numerical Analysis, University of Córdoba,  
Córdoba, Spain.*

<sup>b</sup>*Department of Signal Processing and Communications, Universidad de Alcalá, Madrid,  
Spain.*

<sup>c</sup>*Department of Energy Resource, Iberdrola, Spain.*

---

## Abstract

Wind speed reconstruction is a challenging problem in areas (mainly wind farms) where there are not direct wind measures available. Different approaches have been applied to this reconstruction, such as measure-correlate-predict algorithms, approaches based on physical models such as reanalysis methods, or more recently, indirect measures such as pressure, and its relation to wind speed. This paper adopts the latter method, and deals with wind speed estimation in wind farms from pressure measures, but including different novelties in the problem treatment. Existing synoptic pressure-based indirect approaches for wind speed estimation are based on considering the wind speed as a continuous target variable, estimating then the corresponding wind series of continuous values. However, the exact wind speed is not always needed by wind farms managers, and a general idea of the level of speed is, in the majority of cases, enough to set functional operations for the farm (such as wind turbines stop, for example). Moreover, the accuracy of

---

<sup>☆</sup>This work was supported in part by the Spanish Inter-Ministerial Commission of Science and Technology under Project TIN2011-22794, the Spanish Ministry of Science and Innovation, under project number ECO2010-22065-C03-02, the European Regional Development fund, and the “Junta de Andalucía”, Spain, under Project P08-TIC-3745.

\*Corresponding author.

*Email addresses:* `pagutierrez@uco.es` (P. A. Gutiérrez), `sancho.salcedo@uah.es` (S. Salcedo-Sanz), `chervas@uco.es` (C. Hervás-Martínez), `jsanchezm@uco.es` (J. Sánchez-Monedero)

the models obtained is usually improved for the classification task, given that the problem is simplified. Thus, this paper tackles the problem of wind speed prediction from synoptic pressure patterns by considering wind speed as a discrete variable and, consequently, wind speed prediction as a classification problem, with four wind level categories: low, moderate, high or very high. Moreover, taking into account that these four different classes are associated to four values in an ordinal scale, the problem can be considered as an ordinal regression problem. The performance of several ordinal and nominal classifiers and the improvement achieved by considering the ordering information are evaluated. The results obtained in this paper present the Support Vector Machine as the best tested classifier for this task. *In addition, the use of the intrinsic ordering information of the problem is shown to significantly improve ranks with respect to nominal classification, although differences in accuracy are small.*

*Key words:*

Ordinal classification, ordinal regression, wind speed, pressure patterns, long-term wind speed prediction, wind farms

---

## 1. Introduction

Among renewable energies, wind power is one of the most promising sources of renewable energy in the world, and also the one with a stronger economic impact in developed countries [26]. As an example, wind power installed worldwide by the end of 2009 reaches a total of 157 GW, of which about 76 GW correspond to Europe, and 19 GW only to Spain. Thus, wind power represents over 12% of the total energy consumed in countries such as USA, Germany or Spain, and it is expected that this percentage grows up to an amazing 20% by 2025 [42]. This booming of wind energy has brought together the construction of a huge number of wind farms in the last few years, and, consequently, a good number of new problems associated with the management of these facilities.

Wind speed reconstruction, long-term prediction and wind series analysis are mainly the most important problems faced by wind farm managers in daily operations. These problems are related to different important decisions about the wind farm, such as maintenance stops, production analysis and planning and even micro sitting of new wind turbines. Existing approaches for these problems are mainly based on historic registers of wind measures,

from which statistical models are constructed in order to explain the wind behaviour. These models can be then applied to future values of time in the case of long-term wind speed prediction, or to values in the past in order to reconstruct or analyse and reconstruct wind speed series. Different techniques have been used to obtain these wind speed models, such as statistical methods [28, 45], neural networks [1, 14], support vector machines [34], Bayesian models [32], etc. The majority of the existing techniques used to construct long-term wind speed models are exclusively based on past wind speed data, and some of them include other atmospheric variables as input data, such as local temperature, radiation or pressure at the measuring point. The problem with this approach based on wind measures is that, in some cases, these data are not available, due to fails in the measurement systems, or just because the terrain is a prospective site to install a wind farm, and there is not a meteorological tower installed yet. This problem is even harder in the case of historic analysis or wind series reconstruction, since it is not possible to obtain any direct wind measure if it is not available.

In these problematic cases, the possibility of obtaining indirect measures of wind is currently a hot topic, in which many renewable energy companies are investing lots of resources. In this sense, different recent works have used synoptic pressure<sup>1</sup> as an indirect measure to study different atmospheric phenomenons such as precipitation, pollution or temperature [9, 10, 36, 37, 38, 41, 46]. In the case of the wind, it seems even more evident that a good source of indirect wind measures is the pressure at synoptic scale, since the wind at a given point is a direct function (when the effects of limit boundary layer are removed) of the pressure gradient. Thus, different works have related pressure patterns with local or mesoscale wind [6, 7, 4, 22, 44]. Among them, the work by Hocaoglu et al. [22] has been selected in the experimental section as one of the compared methods, given that it resembles the proposal in this paper in some ways.

Specifically, in this paper, the problem of wind speed estimation in a given point (wind farm), from the corresponding synoptic pressure pattern is tackled. The problem involves daily pressure patterns in a synoptic grid, in this case centred in Spain, and a wind speed module measure. The [main](#)

---

<sup>1</sup>Synoptic scale in meteorology corresponds to atmospheric phenomenons in a horizontal length scale of the order of 1000 kilometres or more. Regarding to synoptic pressure, the majority of high and low-pressure areas that can be seen on weather maps are synoptic-scale systems.

novelty of the paper is that this wind speed is discretized into different levels of wind (classes) in order to treat it as a classification problem. The motivation behind this is that the manager of the wind farm can get enough information from the considered classes in order to set functional operations for the farm (such as wind turbines stop, for example). Note that the exact wind speed value is not usually important for this task. Additionally, higher accuracy can be obtained for a classification task, given that the problem is simplified. Four classes have been considered that cover all the wind speed spectrum of a wind farm operation.

Decision making tasks usually involve that the target variable (wind in this case) takes values in an ordinal scale, what is known as ordinal regression or ordinal classification. This relatively new machine learning field is aimed at finding a prediction rule for ordered categories. Ordinal classification problems arise in statistics [33], and have recently received a lot of attention in the machine learning field, given that it can be applied to a wide range of areas, specially those where a human being can be used to evaluate the target variable [2]: medicine, psychology, some engineering fields, etc. The analysis of the problem reveals that it is different from standard regression, because a distance between the labels cannot be established. And the ordering information is the main difference with respect to nominal classification. To really exploit this order among categories, classifiers should be built including this order in the model formulation, and evaluation measures or metrics should be specifically designed to measure the degree of discrepancy (in the ordinal scale) between the predicted and real categories. All these considerations emphasize the importance of developing and applying ordinal regression models in the field of artificial intelligence.

In this way, ordinal classification can be tackled by adaptations of the well known Support Vector Machine (SVM) classifier [11, 12], the online perceptron algorithm [15], the Proportional Odd Model (POM) [47], by constructing sets of distinct binary classifiers [17] or by transforming them to extended binary problems [5]. Recently, two different algorithms were proposed to generate structured and unstructured monotone ordinal data sets<sup>2</sup> [39]. All these algorithms can provide models to classify data where there

---

<sup>2</sup>Monotonicity is another aspect of ordinal classification, which considers the additional restriction that the predicted labels are monotone with respect to the input decision variables.

exists an order between the different target labels considered, and generally take advantage of this order (instead of simply ignoring it as standard classifiers do).

However, ordinal classification is not usually considered in other fields, and the corresponding real problems are tackled as standard nominal classification (where no order is assumed between the classes). Note that the wind speed characteristics make that the problem can be defined as an ordinal classification problem, in which the different classes (wind speed intervals), can be ordered from the smallest to the largest, in increasing order. In this way, [this paper also makes use of the ordering information for evaluating if better quality classifiers are obtaining](#). The results of ordinal algorithms are compared with respect to nominal classifiers. Five wind farms in Spain are considered, and the main conclusion is that, [although ordinal classification algorithms significantly ranks better than nominal ones, the differences in percentage of correctly classification ratio are quite low](#).

The structure of the rest of the paper is the following: next section presents the definition of the problem. Section 3 presents the main characteristics of the algorithms tested for this problem. The experiments of the paper are then presented in Section 4. Finally, Section 5 closes the paper giving some concluding remarks.

## 2. Problem definition

The problem in this paper may be summarized as follows: Let  $\mathbf{y} = \{y_i, i = 1, \dots, T\}$  be a series of daily wind speed discretized measures at a given point, in such a way that  $y_i \in Y = \{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4\}$ , i.e.  $y$  belongs to one out of 4 classes which are subjected to an ordinal order ( $\mathcal{C}_1 \prec \mathcal{C}_2 \prec \mathcal{C}_3 \prec \mathcal{C}_4$ , where  $\prec$  is an ordering relationship between the labels). In this paper, the different classes for the wind speed have been constructed taking into account the characteristics of the wind turbines, i.e. its power curve. Figure 1 shows the 4 classes established in this paper, which try to model the power curve of the turbines installed in the considered wind farms. Thus,  $\mathcal{C}_1$  contains situations of low wind, where the wind turbine will not produce power,  $\mathcal{C}_2$  summarizes situations in the beginning of the wind power ramp,  $\mathcal{C}_3$  comprises situations in which the production of the wind turbine is significant and  $\mathcal{C}_4$  models situations of high wind speed and power production. Note daily averages of wind speed are studied, so the classes are set to have enough number of samples from each class. Let  $\mathbf{X} = \{\mathbf{x}_i, i = 1, \dots, T\}$  be a series of daily



synoptic-scale pressure measures in a grid. In this case, each component of  $\mathbf{X}$  is a matrix of  $14 \times 13$  surface pressure values (182 values), measured in a grid surrounding the Iberian Peninsula (Figure 2). The problem faced in this paper is a classification problem, consisting of obtaining a machine  $\Phi$  by using a training set  $\{(\mathbf{x}_i, y_i), i = 1, \dots, T_t < T\}$  (the first part of the series), so that for a given value of  $\mathbf{x}_i$ , it estimates the associated value of  $y_i$ , i.e.  $\Phi(\mathbf{x}_i) \rightarrow y_i$ , in such a way that the machine  $\Phi$  minimizes an error measure in an independent test set  $\{(\mathbf{x}_i, y_i), i = T_t + 1, \dots, T\}$  (the rest of the series), to ensure the good generalization of the machine.

Two evaluation metrics have been considered which quantify the accuracy of  $n$  predicted ordinal labels for a given dataset  $\{y_1^*, y_2^*, \dots, y_n^*\}$ , with respect to the true targets  $\{y_1, y_2, \dots, y_n\}$ :

1. Accuracy ( $C$ ) is simply the fraction of correct predictions on individual samples:

$$C = \frac{1}{n} \sum_{i=1}^n I(y_i^* = y_i), \quad (1)$$

where  $I(\cdot)$  is the zero-one loss function and  $n$  is the number of patterns of the dataset.

2. Mean Absolute Error ( $MAE$ ) is the average deviation of the prediction from the true target, i.e.:

$$MAE = \frac{1}{n} \sum_{i=1}^n |\mathcal{O}(y_i^*) - \mathcal{O}(y_i)|, \quad (2)$$

where  $\mathcal{O}(\mathcal{C}_k) = k, 1 \leq k \leq K$ , i.e.  $\mathcal{O}(y_i)$  is the order of class label  $y_i$ .

These measures, commonly found in ordinal regression works [5, 12], are aimed to evaluate two different aspects that can be taken into account when an ordinal regression problem is considered: whether the patterns are generally well classified (accuracy or  $C$ ) and whether the classifier tends to predict a class as close to the real class as possible ( $MAE$ ).

### 3. Evaluated classifiers

The main objective of this paper is to test several methods to tackle wind prediction as a classification problem (as described in Section 2). At the same time, the possible improvement of standard classifiers when including the label ordering information is also evaluated. The classifier description has been

organized in two different groups, nominal classifiers and ordinal classifiers. Support Vector Machine (SVM) methods receive a special attention because they yield the best performance for the problem (as the reader can check out in Section 4).

### 3.1. Nominal classifiers

Very well-known standard nominal classifiers have been taken into account. Their main characteristics are briefly described in the following subsections.

#### 3.1.1. Support Vector Machines

The SVM [3, 13] is perhaps the most common kernel learning method for statistical pattern recognition, widely applied to different real problems [29, 48]. An interesting way of analyzing SVMs [3] is by viewing them as generalized perceptrons with radial basis functions that compute the inner product on transformed input vectors  $\phi(\mathbf{x})$ . These  $\phi(\mathbf{x})$  are denoting feature vectors  $\mathbf{x}$  in a high dimensional Reproducing Kernel Hilbert Space (RKHS), related to  $\mathbf{x}$  by a specific transformation. The reproducing kernel function is used, defined as  $k(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}) \cdot \phi(\mathbf{x}') \rangle$ , where  $\langle \cdot \rangle$  denotes inner product in the RKHS.

The basic idea behind SVMs is to separate the two different classes — they are firstly defined for two classes and then extended to the multiclass case — through a hyperplane which is specified by its normal vector  $\mathbf{w}$  and the bias  $b$ . The hyperplane can be given as  $\langle \mathbf{w} \cdot \phi(\mathbf{x}) \rangle + b = 0$ . SVMs are linear parametric models, based on a linear combination of a kernel function evaluated at the training data points. The nature of the problem makes that their parameters can be obtained as the solution of a convex optimization problem, so there is a single, global optimum. The sparsity is the other characteristic that has made SVM receive a lot of attention: the final number of training points present in the model is a subset of them, known as support vectors. Additionally, hard margins are replaced by soft margins to face non-separable classification sets. This way allows to handle noise, pre-labeling errors and overlapping, which often occur in practice. Slack-variables,  $\xi_i$ , are used to construct a soft margin separating hyperplane [13].

As Vapnik [13] shows, the optimal separating hyperplane is the one which maximizes the distance between the hyperplane and the nearest points of both classes (called margin) and results in the best prediction for unseen data. This can be formulated as a Quadratic Programming (QP) problem.



In order to deal with the multiclass case, a “1-versus-1” approach can be considered, following the recommendations of Hsu and Lin [23]. The idea is to construct a binary classifier per each pair of classes and joining their multiple responses to obtain a final prediction.

### 3.1.2. Other standard nominal classifiers

Apart from the well-known SVM, other standard machine learning classifiers have been considered. This set of classifiers have shown to report good performance in previous machine learning works [30], and they have been selected because they cover some of the more common and accurate approaches for nominal classification (classification trees, boosting ensemble construction, and logistic regression) from those available in the well-known Weka machine learning software [21]. They include:

- The Logistic Model Tree (LMT) classifier [30].
- The C4.5 classification tree inducer [40].
- The AdaBoost.M1 algorithm, using C4.5 as the base learner. AdaBoost, short for Adaptive Boosting, is a machine learning meta-algorithm [18], an algorithm for constructing a “strong” classifier as linear combination of simple “weak” classifiers. The maximum number of iterations has been set to 10 and 100 iterations (Ada10 and Ada100), as done in previous studies [30].
- Multi-logistic regression methods, including the MultiLogistic (MLogistic) and SimpleLogistic (SLogistic) algorithms.
  - MLogistic is an algorithm for building a multinomial logistic regression model, which is one of the more popular approaches for classification. The algorithm includes a ridge estimator to regularize the model and guard against over-fitting [8]. The coefficient matrices is found by a Quasi-Newton Method: the active-sets’ method with the Broyden-Fletcher-Goldfarb-Shanno (BFGS) update.
  - SLogistic is an alternative algorithm to build a multinomial logistic regression model. The process involves using the LogitBoost algorithm [20] to fit *additive logistic regression models* by maximum likelihood. These models are a generalization of the (linear) logistic regression models. This version of the algorithm is

based on controlling the number of variables of the model to avoid over-fitting [30]: an iterative process adds the input variables one by one, and the number of iterations is decided using a cross-validation process.

### 3.2. Ordinal Classifiers

In an ordinal regression problem, the formal definition is the following: an example  $(\mathbf{x}, y)$  is composed of an input vector  $\mathbf{x} \in \mathbb{R}^n$  and an ordinal label  $y_i \in Y = \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_K\}$ . This looks similar to that of a multi-class classification problem, except that the ranks are ordered, so  $\mathcal{C}_1 \prec \mathcal{C}_2 \prec \dots \prec \mathcal{C}_K$  is an additional restriction to the problem.

#### 3.2.1. A Simple Approach to ordinal regression (ASA)

Ordinal information allows ranks to be compared: one could ask “is the rank of  $\mathbf{x}$  greater than  $k$ ?”, considering a fixed rank  $\mathcal{O}(y_k) = k$ . This question is a binary classification problem, and, by asking this question for  $k = 1, 2$ , until  $(K - 1)$ , the rank of a sample  $\mathbf{x}$  can be determined. This is the approach studied in [17], where each binary classification problem is solved independently and the binary probabilistic outputs are transformed to a rank.

#### 3.2.2. Extended Binary Classification (EBC)

Although the approach proposed by Frank and Hall [17] is simple, the generalization performance using the combination step cannot be easily analyzed. The EBC method [31] works differently and allows generalization analysis of the model.

Let us assume that  $f(\mathbf{x}, k)$  is a binary classifier for all the associated questions above. A good prediction would be the following:  $f(\mathbf{x}, k) = 1$  (“yes”) for  $k = 1$  to  $k = y - 1$  (where  $y$  is the rank associated to the pattern  $\mathbf{x}$ ) and  $f(\mathbf{x}, k) = 0$  (“no”) afterwards.

A possible ranking function  $r(\mathbf{x})$  based on all the binary answers  $f(\mathbf{x}, k)$  is the following:

$$r(\mathbf{x}) = 1 + \sum_{k=1}^{K-1} \llbracket f(\mathbf{x}, k) > 0 \rrbracket, \quad (3)$$

being  $\llbracket \cdot \rrbracket$  a Boolean test which is 1 if the inner condition is true, and 0 otherwise. In summary, the EBC method is based on the following three steps:

1. Transform all training samples  $(\mathbf{x}_i, y_i)$  into extended samples  $(\mathbf{x}_i^{(k)}, y_i^{(k)})$ ,  $1 \leq k \leq K - 1$ , using an encoding matrix and weighting patterns with the help of a V-shaped cost matrix.
2. The extended examples are jointly learned by a binary classifier  $f$  with confidence outputs, aiming at a low weighted 0/1 loss.
3. The ranking rule (3) is used to construct a final prediction for new samples.

This framework can be adapted for SVMs, by using a threshold model to estimate  $f(\mathbf{x}, k)$ :

$$f(\mathbf{x}, k) = g(\mathbf{x}) - \theta_k, \quad (4)$$

where  $g(\mathbf{x})$  is a non-linear function defined as  $g(\mathbf{x}) = \langle \mathbf{w} \cdot \phi(\mathbf{x}) \rangle$ , as long as the threshold vector  $\boldsymbol{\theta}$  is ordered, i.e.  $\theta_1 < \theta_2 < \dots < \theta_{K-1}$ . The adaptation of the SVM framework can be performed by simply defining extended kernels. In this paper, the identity matrix is used as the encoding matrix, and the absolute value cost matrix and the standard soft-margin SVM are considered.

### 3.2.3. Gaussian Processes for Ordinal Regression (GPOR)

GPOR [11] is a Bayesian learning algorithm, where the latent variable  $f(\mathbf{x})$  is modelled using Gaussian Processes, and then all the parameters are estimated by using a Bayesian framework. The basic idea is that the values of the latent function  $\{f(\mathbf{x}_i)\}$  are assumed to be the realizations of random variables indexed by their input vectors in a zero-mean Gaussian process. The ideal probability would be:

$$P(y_i | f(\mathbf{x}_i)) = \begin{cases} 1 & \text{if } b_{\mathcal{O}(y_{i-1})} < f(\mathbf{x}) \leq b_{\mathcal{O}(y_i)} \\ 0 & \text{otherwise} \end{cases}.$$

The joint probability of observing the ordinal variables given the latent function is  $P(D|f) = \prod_{i=1}^N P(y_i | f(\mathbf{x}_i))$ , and the Bayes theorem is applied to write the posterior probability  $P(f|D) = \frac{1}{P(D)} \prod_{i=1}^N P(y_i | f(\mathbf{x}_i)) P(f)$ .

In the presence of noise, it is explicitly assumed that the latent functions are contaminated by a Gaussian noise with zero mean and unknown variance  $\sigma^2$ .  $P(f)$  is easily defined as a multivariate Gaussian, by using the fact that the covariance is approximated by kernels. The vector of hyperparameters  $\boldsymbol{\theta}$  includes the width of the Gaussian kernels, the  $\sigma$  for the noise and the set of thresholds.  $P(D)$  or  $P(D|\boldsymbol{\theta})$  is known as the evidence for  $\boldsymbol{\theta}$  and it is

estimated by two different approaches in the paper: a Maximum a Posteriori approach with Laplace approximation and a Expectation Propagation with variational methods.

#### 3.2.4. Support Vector Machines for Ordinal Regression (SVOR).

The previously defined SVM formulation has been adapted to the ordinal regression setting, by simply defining a different threshold  $b_j$  for each class, and specifically adapting the QP problem [43]. Instead of simply deciding the class of the pattern by the sign of the projection  $\mathbf{w}^T \cdot \mathbf{x}$ , the corresponding real line will be split into different intervals by using a threshold vector  $\mathbf{b}$ . This results in parallel hyperplanes with the same  $\mathbf{w}$  and different thresholds  $b_j$ . In this paper, two different implementations for this idea are considered, taken from the work of Chu and Keerthi [12]:

- *SVOR with Explicit constraints* (SVOREX) is based on defining a QP problem where the last set of constraints assuring the order between the thresholds *explicitly* appears in the optimization problem and where the slacks for the  $j$ -th parallel hyperplane are defined for all patterns of class  $j$  and  $j + 1$ .
- *SVOR with Implicit constraints* (SVORIM) is based on redefining again the QP problem, following this principle: instead of considering only the errors from the samples of adjacent categories, samples in all the categories are allowed to contribute errors for each hyperplane. In this way, the ordinal inequalities on the thresholds are *implicitly* satisfied at the optimal solution.

## 4. Experiments

In the following subsections, the description of the datasets and the experimental design is given, together with the description of the methods based on HMMs, which will be used also for comparison purposes. Then, the details on the preprocessing of the datasets are explained, and finally the obtained results with the different considered classifiers are discussed.

### 4.1. Dataset Description and Experimental Design

Five different wind farms have been considered for this study, resulting in five datasets (H, M, P, U and Z). Each dataset includes a series of discretized wind speed values (targets), taken in a tower at 40m of height, and

averaged over 24 hours to obtain daily data values. On the other hand, a series of grids of average daily pressure maps for the same period have been obtained from the National Center for Environmental Prediction/National Center for Atmospheric Research Reanalysis Project (NCEP/NCAR) [27, 35], which are public data profusely used in climatology and meteorology applications. As previously mentioned, an uniform grid in latitude and longitude has been considered, shown in Figure 2, with 182 measurement points, and each element of this grid is one input variable.

For each wind farm, two different sets are obtained, one for training the models and another one for assessing the performance of the algorithms. In this way, the structure of the different datasets used in this study is given in Table 1. The structures of these datasets are challenging, because the distribution of the different classes is clearly imbalanced, with very few situations of high wind speed (class  $\mathcal{C}_4$ ) and lot of patterns belonging to a moderate wind speed class (class  $\mathcal{C}_2$ ).

Since all the tested algorithms are deterministic, they will be run once, deriving a model from the training set and evaluating its accuracy over the test set. Both training and test sets are parts of a wind series, so it is not advisable to do different random partitions of them.

For the selection of the SVM’s hyper-parameters (regularization parameter,  $C$ , and width of the Gaussian functions,  $\gamma$ ), a grid search algorithm was applied with a ten-fold cross-validation, using the following ranges:  $C \in \{10^{-3}, 10^{-2}, \dots, 10^3\}$  and  $\gamma \in \{10^{-3}, 10^{-2}, \dots, 10^3\}$ . This cross-validation has been applied only taking into account the training data, and then repeating the process with the lowest error parameter combination using the complete training set.

#### 4.2. Comparison to Hidden Markov Models

Apart from the methods presented in Section 3, the approach of Hocaoglu et al. [22] based on Hidden Markov Models (HMMs) has been also selected. Although the work has some similarities with the approach presented in this paper (given that wind speed is also estimated from pressure data), some differences have to be outlined. First of all, a complete synoptic grid, with 182 different values ( $14 \times 13$ ) is considered in our approach (see Figure 2). However, the aforementioned paper considered one single atmospheric pressure observation. Pressure and wind speeds values are then quantized in different number of intervals in order to apply discrete HMMs to estimate wind speed. Other important fact is that in [22] hourly wind speed prediction

is considered, whereas in the current approach we manage average daily wind speed values. The lower variability of these daily values can make necessary to use a lower number of states for modelling.

To adapt the approach in [22] to the proposal of this work, a HMM for each wind farm was constructed, considering one single-point pressure value obtained from the 182 values of the grid. Specifically, the absolute value differences between the upper left and the upper right points and between the bottom left and the bottom right ones were averaged. The number of states of each HMM was fixed to 4 states, considering the intervals for wind speed in Figure 1. The observable emissions were considered to be the single-point pressure values, which were discretized in 150 values (in a similar way to [22]). The transition probabilities are obtained and organized in a matrix form in the same way than in [22], as well as the emission matrix.

#### 4.3. Preprocessing of the dataset

As previously stated, the vector of inputs is formed by  $14 \times 13$  surface pressure values (182 values in a grid around the Iberian Peninsula), which results in a very high number of variables. When too many inputs are presented to the standard machine learning algorithms, a very well known problem appears, the *curse of dimensionality*, which can decrease the performance of these algorithms and significantly increase the computational cost. This is not needed for the HMMs described in subsection 4.2, given that only one single pressure value is obtained from the grid to construct the model.

In order to alleviate this problem, a simple approach has been applied, based on the standard technique of Principal Component Analysis (PCA) [25]. PCA is the predominant linear dimensionality reduction technique, and has been widely applied to datasets in all scientific domains. Generally speaking, PCA maps data points from a high dimensional space to a low dimensional space, while keeping all the relevant linear structure intact.

PCA algorithm returns so many principal components (PCs, linear combinations of the input variables) as the total number of inputs, but they are sorted in the following way: the first PC has as high variance as possible (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it will be orthogonal to (i.e. uncorrelated with) the preceding components. Note that it should be decided at a later stage how many PCs are retained when reducing the dimensionality of the problem.



With this aim, the algorithm included in Figure 4 has been applied. The idea is very simple: the coefficients of the PCs are obtained using the training data and all possible combinations from 1 to the number of PCs that retain a 99% of the variance are tested. A 10-fold cross-validation is applied for each combination, estimating the error with one of the simplest existing classifier (a Linear Discriminant Analysis, LDA) in order to limit the computational time. Once the best number of PCs is decided, training and test data are projected into them, and the reduced datasets are returned.

#### 4.4. Results

The results for the two different evaluation measures considered ( $C$  and  $MAE$ , see Equations (1) and (2)) are included in Tables 2 and 3, respectively. Based on the  $C$  and  $MAE$  values, the ranking of each method in each wind farm is obtained ( $R = 1$  for the best performing method and  $R = 12$  for the worst one). The mean accuracy and  $MAE$  ( $\bar{C}$  and  $\bar{M}$ ) as well as the mean ranking ( $\bar{R}_C$  and  $\bar{R}_M$ ) are also included in Tables 2 and 3 ( $\bar{R} = 1$  for the best method and  $\bar{R} = 13$  for the worst one). The first conclusion is that considerably good accuracies are obtained, [what reveals that considering the problem as a classification task can provide an accurate information of the wind farm.](#) Also, the  $MAE$  values are quite low, the algorithms doing a quite good job when ranking the patterns (a  $MAE$  value of 0.2 means that the classifier predictions are, in average, 0.2 categories lower or higher than the target ones).

The approach based on HMMs [22] reports acceptable results but lower in general than those reported by the rest of methods. One possible reason is that the rest of the methods do not take into account the sequential character of wind speed and pressure values, while HMM does. Consequently, it is more difficult for HMMs to improve measures like  $C$  or  $MAE$ , than it is for the rest of more flexible methods.

From these tables, the SVM methods seem to be the most competitive ones from all the different alternatives considered. When analysing the mean ranking and performance, the EBC(SVM) methodology obtains the better results for both measures. The second best methods are SVOREX and SVORIM for  $C$ , and SVORIM for  $MAE$ . Note that high accuracy values can be masking a lower ranking performance (i.e. a high  $MAE$  value), because the classifier can tend to assign rank values far from the real ones.

To determine the statistical significance of the rank differences observed for each method in the different datasets, a non-parametric Friedman test [19]

has been carried out with the  $C$  and  $MAE$  rankings of the different methods (since a previous evaluation of the  $C$  and  $MAE$  values results in rejecting the normality and the equality of variances hypothesis). The test shows that the effect of the method used for classification is statistically significant at a significance level of  $\alpha = 5\%$ , as the confidence interval is  $C_0 = (0, F_{0.05} = 1.96)$  and the F-distribution statistical values are  $F^* = 12.37 \notin C_0$  for  $C$  and  $F^* = 21.67 \notin C_0$  for  $MAE$ . As a result, the test concludes that all algorithms perform statistically differently in mean ranking.

The Bonferroni-Dunn test [16] is an approach to compare all classifiers to a given classifier (a control method), which is more sensitive than comparing all classifiers to each other. This test has been applied to both  $C$  and  $MAE$  rankings using EBC(SVM) as the control method. The test concludes that the differences in  $C$  and  $MAE$  values are significant:

- At a significance level of  $\alpha = 5\%$ , when EBC(SVM) is compared to C4.5, GPOR and HMM using the  $C$  measure (with  $C$  ranking differences of 8.30, 8.10, and 9.30, respectively) and to C4.5, ASA(C4.5), GPOR and HMM using the  $MAE$  measure (with  $MAE$  ranking differences of 8.90, 7.20, 8.80 and 10.40, respectively).
- Additionally, at a significance level of  $\alpha = 10\%$ , when EBC(SVM) is compared to ASA(C4.5) using the  $C$  measure (with  $C$  ranking difference of 6.80), and to Ada10(C4.5) using the  $MAE$  measure (with  $MAE$  ranking difference of 6.60).

It is important to outline that, although the rank differences are significant, the values obtained for the different measures (specially for accuracy,  $C$ ) are very low (see Tables 2 and 3). Consequently, the study cannot clearly establish that the use of the ordering information improves the results obtained by the nominal classifiers. However, it should be mentioned that the number of ordinal methods which obtain better results is higher than in the nominal case, and that the differences for the  $MAE$  measure are generally higher.

## 5. Conclusions

This paper introduced a new approach for daily mean wind speed series estimation, based on synoptic pressure measures. The problem has been stated as a classification task rather than the usual regression approach.

Wind speed was discretized in four different ranges, which gather the main information needed by the experts when managing the wind farm. On the other hand, synoptic pressure measures in a grid have been considered as the input variables. The results of this preliminary study show that the best performing method is the SVM, with very high accuracy and low *MAE* values. Ordering information (more precisely, the EBC and ASA algorithms) do not clearly outperform nominal methods (SVM and C4.5), given that very similar accuracies are obtained (although the differences in ranking over 6 datasets show to be significant).

## References

## References

- [1] T.G. Barbounis and J.B. Theocharis, “Locally recurrent neural networks for long-term wind speed and power prediction,” *Neurocomputing*, vol. 69, no. 4-6, pp. 466-496, 2006.
- [2] R. Bender and U. Grouven, “Ordinal logistic regression in medical research.” *Journal of the Royal College of Physicians of London*, vol. 31, no. 5, pp. 546–51, 2006.
- [3] B. Boser, I. Guyon, and V. Vapnik, “A training algorithm for optimal margin classifiers,” in *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, D. Haussler, Ed. Pittsburgh, PA: ACM Press, 1992, pp. 144–152.
- [4] M. Burlando, M. Antonelli and C. F. Ratto, “Mesoscale wind climate analysis: identification of anemological regions and wind regimes,” *International Journal of Climatology*, vol. 28, pp. 629-641, 2008.
- [5] Cardoso, J.S., da Costa, J.F.P.: Learning to classify ordinal data: The data replication method. *Journal of Machine Learning Research* 8, pp. 1393-1429, 2007.
- [6] L. Carro-Calvo, S. Salcedo-Sanz, N. Kirchner-Bossi, A. Portilla-Figueras, L. Prietoc, R. Garcia-Herrera and E. Hernández-Martín, “Extraction of synoptic pressure patterns for long-term wind speed estimation in wind farms using evolutionary computing,” *Energy*, vol. 36, pp. 1571-1581, 2011.

- [7] L. Carro-Calvo, S. Salcedo-Sanz, L. Prieto, N. Kirchner-Bossi, A. Portilla-Figueras and S. Jiménez-Fernández, “Wind speed reconstruction from synoptic pressure patterns using an evolutionary algorithm,” *Applied Energy*, vol. 89, pp. 347-354, 2012.
- [8] S. le Cessie and J. van Houwelingen, “Ridge estimators in logistic regression,” *Applied Statistics*, vol. 41, no. 1, pp. 191-201, 1992.
- [9] Z.H. Chen, S.Y. Cheng, J.B. Li, X.R. Guo, W.H. Wang and D.S. Chen, “Relationship between atmospheric pollution processes and synoptic pressure patterns in northern China,” *Atmospheric Environment*, vol. 42, no. 24, pp. 6078-6087, 2008.
- [10] W. L. Cheng “Synoptic weather patterns and their relationship to high ozone concentrations in the Taichung Basin,” *Atmospheric Environment*, vol. 35, no. 29, pp. 4971-4994, 2001.
- [11] W. Chu and Z. Ghahramani, “Gaussian processes for ordinal regression,” *Journal of Machine Learning Research*, vol. 6, pp. 1019-1041, 2005.
- [12] W. Chu and S. S. Keerthi, “Support vector ordinal regression,” *Neural Computation*, vol. 19, no. 3, pp. 792-815, 2007.
- [13] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [14] A. Costa, A. Crespo, J. Navarro, G. Lizcano, H. Madsen and E. Feitosa, “A review on the young history of the wind power short-term prediction,” *Renewable and Sustainable Energy Reviews*, vol. 12, pp. 1725-1744, 2008.
- [15] K. Crammer and Y. Singer, “Online ranking by projecting,” *Neural Computation*, vol. 17, no. 1, pp 145-175, 2005.
- [16] J. Demar, “Statistical comparisons of classifiers over multiple data sets,” *Journal of Machine Learning Research*, vol. 7, pp. 1-30, 2006.
- [17] E. Frank and M. Hall, “A simple approach to ordinal classification,” in *EMCL 2001: Proceedings of the 12th European Conference on Machine Learning*, London, UK: Springer-Verlag, 2001, pp. 145-156

- [18] Y. Freund and R. E. Schapire, “Experiments with a new boosting algorithm,” in *Proceedings of the Thirteenth International Conference on Machine Learning*, Morgan Kaufmann, pp. 148-156, 1996.
- [19] M. Friedman, “A comparison of alternative tests of significance for the problem of  $m$  rankings,” *Annals of Mathematical Statistics*, vol. 11, no. 1, pp. 86-92, 1940.
- [20] J. Friedman, T. Hastie, and R. Tibshirani, “Additive logistic regression: a statistical view of boosting,” *The Annals of Statistics*, vol. 38, no. 2, pp. 337-374, 2000.
- [21] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The weka data mining software: an update,” *SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.
- [22] F. O. Hocaoglu, Ö. N. Gerek, M. Kurban, “A novel wind speed modeling approach using atmospheric pressure observations and hidden Markov models,” *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 98, no. 8-9, pp. 472-481, 2010.
- [23] C.-W. Hsu and C.-J. Lin, “A comparison of methods for multi-class support vector machines,” *IEEE Transaction on Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002.
- [24] P. A. Jiménez, J. F. González-Rouco, J. P. Montavez, E. García-Bustamante and J. Navarro, “Climatology of wind patterns in the north-east of the Iberian Peninsula,” *International Journal of Climatology*, vol. 29, pp. 501-525, 2009.
- [25] I. T. Jolliffe, *Principal Component Analysis*, 2nd ed. Springer, Oct. 2002.
- [26] J. K. Kaldellis and D. Zafirakis “The wind energy (r)evolution: A short review of a long history,” *Renewable Energy*, vol. 36, no. 7, pp. 1887-1901, 2011.
- [27] E. Kalnay et al. “The NCEP/NCAR reanalysis project,” *Bulletin of the American Meteorological Society*, vol. 77, pp. 437-471, 1996.
- [28] M. Khashei, M. Bijari and G. Raissi-Ardali, “Improvement of Auto-Regressive Integrated Moving Average Models Using Fuzzy Logic and

- Artificial Neural Networks (ANNs),” *Neurocomputing*, vol. 72, no. 4-6, pp. 956-967, 2009.
- [29] O. Kisi, and M. Cimen, “Precipitation forecasting by using wavelet-support vector machine conjunction model,” *Engineering Applications of Artificial Intelligence*, in press, 2011.
  - [30] N. Landwehr, M. Hall, and E. Frank, “Logistic model trees,” *Machine Learning*, vol. 59, no. 1, pp. 161-205, 2005.
  - [31] L. Li and H.-T. Lin, “Ordinal regression by extended binary classification,” in *Advances in Neural Information Processing Systems*, vol. 19, pp. 865-872, 2007.
  - [32] G. Li, J. Shi, J. Zhou “Bayesian adaptive combination of short-term wind speed forecasts from neural network models,” *Renewable Energy*, vol. 36, no. 1, pp. 352-359, 2011.
  - [33] P. McCullagh, J. A. Nelder, “Generalized Linear Models,” *Monographs on Statistics and Applied Probability*, Chapman & Hall/CRC, 2nd edition, 1989.
  - [34] M. A. Mohandes, T. O. Halawani, S. Rehman and A. A. Hussain, “Support vector machines for wind speed prediction,” *Renewable Energy*, vol. 29, pp. 939-947, 2004.
  - [35] <http://www.esrl.noaa.gov/psd/data/reanalysis/reanalysis.shtml>
  - [36] S. Osowski, and K. Garanty, “Forecasting of the daily meteorological pollution using wavelets and support vector machine,” *Engineering Applications of Artificial Intelligence*, vol. 20, no. 6, pp. 745-755, 2006.
  - [37] A. Paniagua-Tineo, S. Salcedo-Sanz, C. Casanova-Mateo, E.G. Ortiz-García, M.A. Cony, E. Hernández-Martín, “Prediction of daily maximum temperature using a support vector regression algorithm,” *Renewable Energy*, vol. 36, no. 11, pp. 3054-3060, 2011.
  - [38] D. Paredes, D., R. M. Trigo, R. García-Herrera and I. F. Trigo, “Understanding precipitation changes in Iberia in early spring: weather typing and storm tracking approaches,” *Journal of Hydrometeorology*, vol. 7, pp. 101-113, 2006.



- [39] R. Potharst, A. Ben-David, and M. van Wezel, “Two algorithms for generating structured and unstructured monotone ordinal data sets,” *Engineering Applications of Artificial Intelligence*, vol. 22(4-5), pp. 491-496, 2009.
- [40] J. R. Quinlan, C4.5: programs for machine learning. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.
- [41] R. Romero, G. Summer, C. Ramis and A. Genoves, “A classification of the atmospheric circulation patterns producing significant daily rainfall in the Spanish Mediterranean area,” *International Journal of Climatology*, vol. 19, 765-785, 1999.
- [42] R. Saidur, M.R. Islam, N.A. Rahim, K.H. Solangi, “A review on global wind energy policy,” *Renewable and Sustainable Energy Reviews*, vol. 14, no. 7, pp. 1744-1762, 2010.
- [43] A. Shashua and A. Levin, “Ranking with large margin principle: Two approaches,” *Advances in Neural Information Processing Systems*, vol. 15, pp. 937-944, MIT Press, Cambridge, 2003.
- [44] C. Soriano, A. Fernández and J. Martin-Vide, “Objective Synoptic classification combined with high resolution meteorological models for wind mesoscale studies,” *Meteorology and Atmospheric Physics*, vol. 91, pp. 165-181, 2006.
- [45] J. L. Torres, A. García, M. De Blas and A. De Francisco, “Forecast of hourly average wind speed with ARMA models in Navarre (Spain),” *Solar Energy*, vol. 79, pp. 65-77, 2005.
- [46] R. M. Trigo and C. C. DaCamara, “Circulation weather types and their influence on the precipitation regime in Portugal,” *International Journal of Climatology*, vol. 20, pp. 1559-1581, 2000.
- [47] J. Verwaeren, W. Waegeman and B. D. Baets, “Learning partial ordinal class memberships with kernel-based proportional odds models,” *Computational Statistics & Data Analysis*, in press (doi:10.1016/j.csda.2010.12.007), 2011.

- [48] A. Zarnani, P. Musilek, X. Shi, X. Ke, H. He, and R. Greiner, “Learning to predict ice accretion on electric power lines,” *Engineering Applications of Artificial Intelligence*, vol. 25, no. 3, pp. 609-617, 2012.

## List of Tables

1	Structure of training and test sets: total number of patterns (Size), number of pattern in each class ( $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4$ ) (Distribution) and final number of Principal Components (PCs) . . . . .	24
2	Test accuracy ( $C(\%)$ ) results obtained by using the different methods evaluated . . . . .	24
3	Test Mean Absolute Error ( $MAE$ ) results obtained by using the different methods evaluated . . . . .	25

## List of Figures

1	Wind speed classes ( $\mathcal{C}_1 \prec \mathcal{C}_2 \prec \mathcal{C}_3 \prec \mathcal{C}_4$ ) and its relationship with the power curve of the wind turbines. . . . .	26
2	Synoptic pressure grid considered (Sea Level Pressure values have been used in this paper). . . . .	26
3	Location of the wind farms considered in this work. . . . .	27
4	Algorithm for deciding the number of principal components . .	27

Table 1: Structure of training and test sets: total number of patterns (Size), number of pattern in each class ( $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4$ ) (Distribution) and final number of Principal Components (PCs)

Wind farm	Training		Test		PCs <sup>1</sup>
	Size	Distribution	Size	Distribution	
H	2196	(416,1478,272,30)	1098	(200,790,99,9)	13
M	2231	(220,1590,396,52)	1115	(173,779,147,16)	10
P	2185	(773,1076,295,41)	1092	(409,538,125,20)	11
U	2017	(527,1167,280,43)	1008	(361,547,85,15)	6
Z	1749	(901,637,184,27)	874	(516,279,68,11)	13

<sup>1</sup>: This value has been obtained using the algorithm in Figure 4.

Table 2: Test accuracy ( $C(\%)$ ) results obtained by using the different methods evaluated

Classifier	Wind farm					$\bar{C}(\%)$	$\bar{R}_C$
	H	M	P	U	Z		
SVM	<b>75.77</b>	<i>73.32</i>	63.64	62.50	<b>70.48</b>	69.14	3.10
LMT	75.05	70.94	56.23	62.80	64.65	65.93	6.10
C45	67.67	70.04	50.37	57.54	55.72	60.27	10.60
Ada10(C45)	69.22	68.43	53.66	<b>63.39</b>	61.10	63.16	8.40
Ada100(C45)	74.23	72.20	60.07	62.50	66.36	67.07	5.80
MLogistic	74.50	71.66	52.75	57.44	62.24	63.72	8.00
SLogistic	75.05	71.21	54.03	57.34	62.36	64.00	7.60
ASA(C45)	70.86	70.76	56.41	57.34	56.75	62.42	9.30
EBC(SVM)	<b>75.77</b>	<b>73.90</b>	63.64	62.50	<b>70.48</b>	<b>69.26</b>	<b>2.70</b>
SVOREX	<i>75.50</i>	73.27	<b>64.65</b>	<i>62.90</i>	68.99	69.06	<i>2.90</i>
SVORIM	75.23	73.36	<i>64.56</i>	<i>62.90</i>	<i>69.34</i>	<i>69.08</i>	<b>2.70</b>
GPOR	71.95	69.87	49.27	54.27	59.04	60.88	10.80
HMM	71.13	69.42	42.49	51.39	56.29	58.14	12.00

The best result is in bold face and the second best result in italics

Table 3: Test Mean Absolute Error ( $MAE$ ) results obtained by using the different methods evaluated

Classifier	Wind farm					$\bar{M}$	$\bar{R}_M$
	H	M	P	U	Z		
SVM	<b>0.242</b>	<i>0.267</i>	0.365	0.382	<i>0.300</i>	0.311	2.90
LMT	0.250	0.293	0.459	0.383	0.373	0.352	6.20
C45	0.335	0.310	0.540	0.434	0.487	0.421	10.90
Ada10(C45)	0.314	0.318	0.492	<i>0.381</i>	0.420	0.385	8.60
Ada100(C45)	0.260	0.281	0.419	0.389	0.354	0.341	6.00
MLogistic	0.258	0.288	0.514	0.433	0.405	0.379	7.80
SLogistic	0.250	0.293	0.495	0.434	0.400	0.374	7.70
ASA(C45)	0.293	0.299	0.465	0.438	0.463	0.392	9.40
EBC(SVM)	<b>0.242</b>	<b>0.261</b>	0.364	0.382	<b>0.295</b>	<b>0.309</b>	<b>2.20</b>
SVOREX	<i>0.245</i>	0.268	<b>0.354</b>	<b>0.378</b>	0.317	<i>0.312</i>	2.70
SVORIM	0.248	<i>0.267</i>	<i>0.355</i>	<b>0.378</b>	0.314	<i>0.312</i>	<i>2.60</i>
GPOR	0.289	0.316	0.526	0.472	0.513	0.423	11.00
HMM	0.301	0.322	0.646	0.525	0.535	0.466	12.60

The best result is in bold face and the second best result in italics



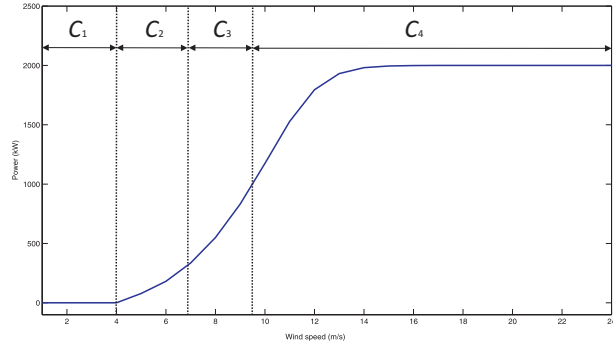


Figure 1: Wind speed classes ( $C_1 \prec C_2 \prec C_3 \prec C_4$ ) and its relationship with the power curve of the wind turbines.

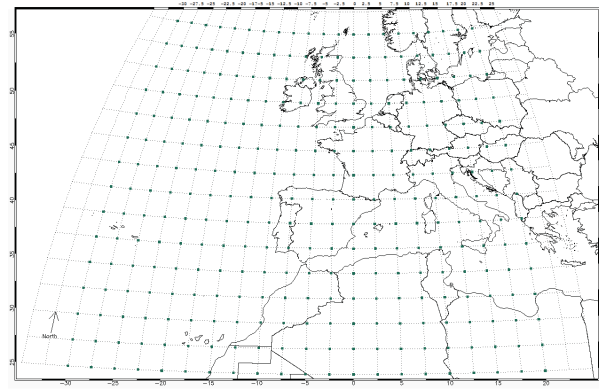


Figure 2: Synoptic pressure grid considered (Sea Level Pressure values have been used in this paper).

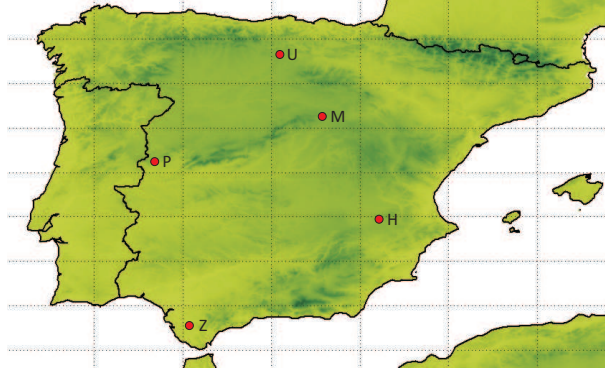


Figure 3: Location of the wind farms considered in this work.

#### **Deciding # of Principal Components::**

**Require:** Training dataset ( $Tr$ ), Test dataset ( $Te$ )

**Ensure:** Projected training dataset ( $Tr^*$ ), Projected test dataset ( $Te^*$ )

- 1: Apply PCA to  $Tr$ , without considering  $Te$
- 2:  $Max \leftarrow$  Number of PCs retaining a 99% of the total variance of the dataset
- 3: **for**  $i = 1 \rightarrow Max$  **do**
- 4:    $Tr_i \leftarrow Tr$  projected over the  $i$  first PCs.
- 5:   Apply a ten-fold cross-validation method, considering  $Tr_i$  data and the LDA classifier.
- 6:    $e_i \leftarrow$  cross-validated error of the classifier.
- 7: **end for**
- 8:  $n \leftarrow \text{argmin}_i e_i$
- 9:  $Tr^* \leftarrow Tr$  projected over the  $n$  first PCs.
- 10:  $Te^* \leftarrow Te$  projected over the  $n$  first PCs.
- 11: **return**  $Tr^*$  and  $Te^*$

Figure 4: Algorithm for deciding the number of principal components